UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/666,425	09/19/2003	**David E. Martin**	2002P15891US01	8309

7590 10/27/2008

Siemens Corporation
Intellectual Property Department
170 Wood Avenue South
Iselin, NJ 08830

EXAMINER	
VERDI, KIMBLEANN C	
ART UNIT	PAPER NUMBER
2194	

MAIL DATE	DELIVERY MODE
10/27/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/666,425	MARTIN, DAVID E.	
	Examiner	Art Unit	
	KimbleAnn Verdi	2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *July 22, 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-38* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-38* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claims 1-38 are pending in the current application.

Claim Rejections - 35 USC § 112

1. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

2. Claims 1-38 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

3. Claim 1 lines 8-9, claim 20, lines 9-10, and claim 38 lines 9-11, it is not clearly understood if the request is used to request an extension to be loaded or is used to request the execution of the application. Is the request for the extension to be loaded sent to the application which causes the execution of the application? (i.e. what is the relationship or connection between request an extension to be loaded and requesting execution of the application).

4. Claim 1 lines 11-13, claim 17 lines 7-10, claim 20, lines 12-14, claim 33, lines 10-12, and claim 38 lines 14-16, it is not clearly understood how the first mode and second mode are adapted to generate Dynamic Link Library projects. How does the first mode and second mode generate Dynamic Link Library projects? And Are the Dynamic Link Library projects created or loaded by the first mode and second mode? (i.e. what is the

first mode and second mode and how do they generate a Dynamic Link Library project, and what is the relationship or connection between an extension to be loaded and generating a Dynamic Link Library project).

5. Claim 1, lines 5-6 and claim 33, lines 6-7, it is not clearly understood who is converting the specific formatted load application request message. Is the ODK subsystem or PLC responsible for converting the specific formatted load application request message? (i.e. who is converting the specific formatted load application request message).

6. Claim 5, lines 3-4, it is not clearly understood how the helper class is adapted to access Step 7 type data from a buffer. How does the helper class access Step 7 type data from a buffer? (i.e. what is the helper class and how does it access Step 7 type data from a buffer).

7. Claim 17, line 3, it is not clearly understood how the one or more extensions are adapted for use in a real-time operating environment. How are the one or more extensions created, changed, or modified for use in a real-time operating environment? (i.e. what is the characteristic in the one or more extensions which allows for use in a real-time operating environment).

8. Claim 33, lines 7-8, it is not clearly understood who is sending the generic formatted message to an application. Is the ODK subsystem or PLC responsible for sending the generic formatted message to an application? (i.e. who is sending the generic formatted message to an application).

Claim Rejections - 35 USC § 101

9. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

10. Claim 38 rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

11. Claim 38 is directed to non-statutory subject matter. In view of Applicant's disclosure, paragraph [0051], the medium is not limited to storage medium embodiments, instead being defined as including both storage medium embodiments (e.g., CD-ROM discs, ROM cards, floppy discs, magnetic tapes, computer hard drives) and transmission medium embodiments (e.g., carrier waves). As such, the claim is not limited to statutory subject matter and is therefore non-statutory. To overcome this type of 101 rejection the claims need to be amended to include only the physical computer media (e.g., storage media) and not a transmission media or other intangible or non-functional media. Examiner suggests the recitation of line 1 "computer usable medium", to recite "computer storage medium.

Claim Rejections - 35 USC § 103

12. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

13. Claims 1-16 and 20-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Papadopoulos et al. (hereinafter Papadopoulos) (U.S. Patent 6,282,454) in view of Hammond (U.S. Patent 5,974,470) and further in view of Kodosky et al. (hereinafter Kodosky) (U.S. Patent 6,173,438 B1).

14. As to claim 1, Papadopoulos teaches the invention substantially as claimed including a method comprising the steps of:

sending a specific formatted load application request message (e.g. request sent from PLC 32, to back plane driver 56 of web server 30, Fig. 3) from a programmable logic controller (PLC) to an open development kit (ODK) subsystem (e.g. web server 30 or any other interface to translate PLC 32, Fig. 3, request message to different format, back plane driver 56, Fig. 3 receives request from PLC's ladder logic application 36, Fig.2, col. 4, lines 36-39);

converting the specific formatted load application request message (e.g. client task request message) to a generic formatted load application request message (e.g. converted to HTTP message) (the server 20, Fig. 1, acts as the HTTP interpreter

through TCP/IP stack 24, to interact with network interface 16 and application program 22 of PLC, Fig. 1, col. 3, lines 50-56);

sending the generic formatted load application request message from the ODK subsystem (e.g. web server) to an application (e.g. web browser) (the TCP/IP stack 24, enables data transfers between the application 22 and the user 2 through the internet 14, Fig. 1, utilizing IP protocol, col. 3, lines 53-60).

15. Papadopoulos does not explicitly disclose requesting an extension to be loaded; and requesting execution of the application, the application comprising an Application Wizard adapted to provide a first mode and a second mode, the first mode adapted to generate a Real Time Dynamic Link Library that is usable in a deterministic environment with fixed scan cycles, the second mode adapted to generate a Dynamic Link Library that is usable in a non-deterministic environment with non-fixed scan cycles.

16. However Hammond teaches requesting an extension (e.g. DLL) to be loaded (system receives request for loading DLL in Windows™ operating system, col. 2, lines 36-39 of Hammond) and requesting execution of the application (col. 2, lines 39-49 of Hammond).

17. It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified the PLC's Application Program request of Papadopoulos with the teachings of requesting an extension (e.g. DLL) to be loaded

from Hammond because this feature would have provided any application access to a set of autonomous functions which could be linked to the application at run time per application request (col. 1, lines 30-34 and col. 5, 64-66 of Hammond).

18. In addition, Kodosky teaches the application comprising an Application Wizard adapted to provide a first mode and a second mode (col. 22, lines 39-54), the first mode adapted to generate a Real Time Dynamic Link Library that is usable in a deterministic environment with fixed scan cycles (co. 22, lines 41-43), the second mode adapted to generate a Dynamic Link Library that is usable in a non-deterministic environment with non-fixed scan cycles (col. 22, lines 41-45).

19. It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have further modified the DLL of Papadopoulos as modified by Hammond with the teachings of a DLL from Kodosky because this feature would have further provided a mechanism for an embedded application to load DLLs and to invoke or call functions in DLLs (col. 22, lines 39-41).

20. As to claim 2, Papadopoulos as further modified teaches the steps of:

initializing callback function pointers for use by the application as callback functions into the ODK subsystem (client application queues callback function associated with the request, col. 5, lines 49-51 of Papadopoulos); and

initializing (e.g. linking to application) the extension (e.g. DLL) after the extension (e.g. DLL) is loaded (DLL may be loaded and linked to an application at run time, col. 1, lines 31-32 of Hammond).

21. As to claim 3, Papadopoulos as further modified teaches wherein the extension is a dynamic load library (DLL) (DLL consists of functions any application can use (e.g. extends the application functionality), col. 1, lines 33-34 of Hammond).

22. As to claim 4, Papadopoulos as further modified teaches the step of checking whether a stop to run transition has occurred in the PLC (user presses start box 168, Fig. 4, ladder logic diagram, on browser screen, col. 10, lines 8-12 of Papadopoulos) and if so, sending a specific formatted activate application message from the PLC to the ODK subsystem (client task 58, allows an application to receive a new ladder logic MSTR request via the back plane driver 56, Fig. 3, col. 5, lines 36-38 of Papadopoulos).

23. As to claim 5, Papadopoulos as further modified teaches the step of calling an activate function in the application by the ODK subsystem (client task 58, allows an application to receive a new ladder logic MSTR request (e.g. activate function) via the back plane driver 56, Fig. 3, col. 5, lines 36-38 of Papadopoulos) thereby permitting scan cycle execution (e.g. test application executed, overload relay detects overload condition, col. 9, line 47 of Papadopoulos), the Real Time Dynamic Link Library project comprising a helper class adapted to access Step 7 type data from a buffer that is

passed between a central processing unit and the application (col. 18, lines 10-21 of Kodosky).

24. As to claim 6, Papadopoulos as further modified wherein the requesting execution step includes checking whether there are requests for application execution in the PLC, (back plane driver detects new MSTR block request, col. 5, lines 51-52 of Papadopoulos) and if so, sending a specific formatted execution request (e.g. request sent from PLC to back plane of web server) from the PLC to the ODK subsystem (MSTR functions allow programs running in PLC 32, Fig. 3, to send commands (e.g. messages) to remote node on TCP/IP network (e.g. website), col. 5, lines 22-26, back plane driver 56, Fig. 3 receives request from PLC's ladder logic application 40, Fig.2, col. 4, lines 36-39 of Papadopoulos), converting the specific formatted execution request to a generic execution request message (e.g. converted to HTTP message) (the server 20, Fig. 1, acts as the HTTP interpreter through TCP/IP stack 24, to interact with network interface 16 and application program 22 of PLC, Fig. 1, col. 3, lines 50-56 of Papadopoulos), and sending a generic execution request from the ODK subsystem to the application (the TCP/IP stack 24, enables data transfers between the application 22 and the user 2 through the internet 14, Fig. 1, utilizing IP protocol, col. 3, lines 53-60 of Papadopoulos), and wherein the deterministic environment is a WinAC RTX Environment (col. 12, lines 34-39 and col. 18, lines 22-32 of Kodosky).

25. As to claim 7, Papadopoulos as further modified teaches the step of executing the generic execution request by the application (col. 22, lines 61-63 of Kodosky).

26. As to claim 8, Papadopoulos as further modified teaches the steps of:

sending a generic response from the application to the ODK subsystem (TCP/IP stack receives messages (e.g. response message from application) over Ethernet, col. 4, lines 55-57, and enables data transfer (e.g. communication) between the user 2 and the network interface 16, Fig. 1 over the internet, col. 3, lines 57-58 of Papadopoulos);

converting the generic response to a specific formatted response (TCP/IP stack 54 returns a response to the client task 58, Fig. 3, col. 6, lines 50-51 of Papadopoulos); and

sending the specific formatted response from the ODK subsystem to the PLC (client task 58, passes response to the back plane driver 56, Fig. 3, col. 6, lines 51-52, when response received back plane driver 56, Fig. 3passes it back to the PLC MSTR blocks, col. 5, lines 31-33 of Papadopoulos).

27. As to claim 9, of Papadopoulos as further modified teaches the step of returning at least one of data and a control block from the application to the ODK subsystem (enables data transfer between the user 2, Fig. 1 and the website 4, Fig. 1, over the internet, col. 3, lines 57-58, the application calls a routine to pass the MSTR (blocks) response to the driver, col. 5, lines 57-59 of Papadopoulos), and from the ODK

subsystem to the PLC (the driver passes back the response (MSTR blocks from above) to the ladder logic program of the PLC, col. 5, lines 29-30 and 59-61 of Papadopoulos).

28. As to claim 10, Papadopoulos as further modified teaches checking whether any requests for information are waiting in the application (Ethernet driver 46 places receive buffers (e.g. contain received messages from application) in the receive queue, on interrupt Ethernet driver 46 examines receive queue and if there are messages passes the receive queue to the TCP/IP stack 54, Fig. 3, col. 5, lines 4-6 of Papadopoulos), and, if so, requesting information from the PLC by the application (the server task 60, Fig. 3 allows an application to issue a request command to the PLC's executive program and receive a response (e.g. information), col. 5, lines 34-36 of Papadopoulos);

executing a function in the ODK subsystem specified by the application (based on the TCP/IP event (e.g. message received on TCP/IP stack from application), the server task 60, Fig. 3, uses the connection machine and transaction machine to advance the transaction (e.g. function), col. 7, lines 65-67 of Papadopoulos); and

performing a task in the PLC associated with the executed function in the ODK subsystem (server task 60 posts requests to the back plane driver 56, Fig. 3, col. 7, lines 42-43; the server task 60, Fig. 3 allows an application to issue a request command to the PLC's executive program and receive a response (e.g. information), col. 5, lines 34-36 of Papadopoulos).

29. As to claim 11, Papadopoulos as modified teaches wherein in the requesting information step the application uses a call back pointer to generically request information (application queues both the request and call back function associated with the request, col. 5, lines 41-42 of Papadopoulos) and the executing step executes the function in the ODK subsystem corresponding to the callback pointer (when back plane driver 56, Fig. 3 services the request it calls the associated call back function, col. 5, lines 42-43 of Papadopoulos).

30. As to claim 12, Papadopoulos further as modified teaches in the executing a function step (call back function calls operating routine to pass message, col. 5, lines 45-46 of Papadopoulos), the function is provided by a dynamic link library (DLL for Windows™ operating system contains functions for user interface tasks of message sending, col. 1, lines 44-50 of Hammond).

31. As to claim 13, Papadopoulos as further modified teaches the step of returning a specific formatted response from the PLC to the ODK subsystem (server task 60 posts requests (to PLC) to back plane driver 56, and an associated call back routine sends the response to the server task 60, Fig. 3, col. 7, lines 42-45 of Papadopoulos), the ODK subsystem converting the specific formatted response to a generic response (when the response message is received the server task 60, Fig. 3, finds connection and transaction machine in order to send response, col. 8 lines 3-5 of Papadopoulos), and returning the generic response from the ODK subsystem to the application (server

task 60 uses TCP/IP stack 54 to transmit message, Fig. 3, col. 8, lines 6-7 of Papadopoulos), the ODK subsystem comprising an ODK RTX Proxy DLL usable in the deterministic environment (col. 22, lines 39-45 of Kodosky), the ODK RTX Proxy DLL adapted to communicate through shared memory with an ODK SB Add-on DLL in the non-deterministic environment (col. 22, lines 45-49, and col. 18, lines 5-6 and 19-21 of Kodosky).

32. As to claim 14, Papadopoulos as further modified teaches wherein when the checking

determines that there are no requests for information waiting, and further comprising:

waiting until the PLC transitions from a run state to a stop state (user presses Stop box 170, Fig. 4, ladder logic diagram, on browser screen, col. 10, lines 8-12 of Papadopoulos);

sending a deactivate request from the PLC to the ODK subsystem (client task 58, allows an application to receive a new ladder logic MSTR request via the back plane driver 56, Fig. 3, col. 5, lines 36-38 of Papadopoulos); and

calling a deactivate function in the application (client task 58, allows an application to receive a new ladder logic MSTR request (e.g. deactivate function) via the back plane driver 56, Fig. 3, col. 5, lines 36-38 of Papadopoulos).

33. As to claim 15, Papadopoulos as further modified teaches wherein when a memory clear or PLC shutdown occurs (e.g. PLC stops) (user presses Stop box 170,

Fig. 4, ladder logic diagram, on browser screen, col. 10, lines 8-12 of Papadopoulos), the step of calling a release function in the application (client task 58, allows an application to receive a new ladder logic MSTR request (e.g. release function) via the back plane driver 56, Fig. 3, col. 5, lines 36-38 of Papadopoulos) and unloading the extension (DLL) occurs (free DLL module associated with application, col. 8, lines 26-27, Free Module function removes mapping of DLL to application, steps 78 and 82, Fig. 4 of Hammond) .

34. As to claim 16 Papadopoulos as further modified teaches wherein in the sending a load application request from a PLC is one of a soft PLC (e.g. ladder logic application of PLC) (back plane driver 56, Fig. 3 receives request from PLC's ladder logic application 40, Fig.2, col. 4, lines 36-39, PLC application program includes ladder logic program for controlling I/O devices 40, Fig. 2, col. 4, lines 37-39 of Papadopoulos).

35. As to claims 20-21, these claims are rejected for the same reasons as claims 1-2 respectively, since claims 20-21 recite the same or equivalent invention, see the rejections to claims 1-2 above.

36. As to claims 22-30, these claims are rejected for the same reasons as claims 4-12 respectively, since claims 22-30 recite the same or equivalent invention, see the rejections to claims 4-12 above.

37. As to claim 31, this claim is rejected for the same reasons as claim 16 since claim 31 recites the same or equivalent invention, see the rejection to claim 16 above.

38. As to claim 32, this claim is rejected for the same reasons as claim 13 since claim 32 recites the same or equivalent invention, see the rejection to claim 13 above.

39. As to claim 33, this claim is rejected for the same reasons as claim 1 since claim 33 recites the same or equivalent invention, see the rejection to claim 1 above.

40. As to claim 34, this claim is rejected for the same reasons as claim 8 since claim 34 recites the same or equivalent invention, see the rejection to claim 8 above.

41. As to claim 35, Papadopoulos as further modified teaches wherein at least one of the means includes using a dynamic link library that loads replaceable functionality (upgrade DLL to new code 24, compile/link to create new DLL 28, Fig. 4, of Shipley and col. 3, lines 55-56, of Hammond).

42. As to claim 36, Papadopoulos as further modified teaches wherein the generic formatted message is one of a response message (TCP/IP stack receives messages (e.g. response message from application) over Ethernet, col. 4, lines 55-57, and enables data transfer (e.g. communication) between the user 2 and the network

interface 16, Fig. 1 over the internet, col. 3, lines 57-58 of Papadopoulos) and a request for information message (the server task 60, Fig. 3 allows an application to issue a request command to the PLC's executive program and receive a response (e.g. information), col. 5, lines 34-36 of Papadopoulos).

43. As to claim 37, this claim is rejected for the same reasons as claim 16 since claim 37 recites the same or equivalent invention, see the rejection to claim 16 above.

44. As to claim 38, this claim is rejected for the same reasons as claim 1 since claim 38 recites the same or equivalent invention, see the rejection to claim 1 above.

45. Claim 17 is rejected under 35 U.S.C. 103(a) as being unpatentable over Papadopoulos et al. (hereinafter Papadopoulos) (U.S. Patent 6,282,454) in view of Meek et al. (hereinafter Meek) (U.S. Patent 6,121,924), and further in view of Kodosky et al. (hereinafter Kodosky) (U.S. Patent 6,173,438 B1).

46. As to claim 17, Papadopoulos teaches the invention substantially as claimed including a system for open development, comprising:

one or more extensions (e.g. client task 58, Fig. 3) adapted for use in a real-time operating environment (real time operating system 44, Fig. 3);

and a CPU adapted to execute a programmable logic controller (PLC) application program in the real-time operating environment and adapted to execute the

one or more extensions, wherein the one or more extensions (e.g. client task 58, Fig. 3) provide access into the scan cycle of the PLC (client task 58, allows an application to receive a new ladder logic MSTR request (e.g. activate function) via the back plane driver 56, Fig. 3, col. 5, lines 36-38, for example: test application executed, overload relay detects overload condition, col. 9, line 47 of Papadopoulos).

47. Papadopoulos does not explicitly disclose a virtual CPU adapted to provide replaceable functionality to the operation of the PLC; and

the system comprising an Application Wizard adapted to provide a first mode and a second mode, the first mode adapted to generate a Real Time Dynamic Link Library that is usable in a deterministic environment with fixed scan cycles, the second mode adapted to generate a Dynamic Link Library that is usable in a non-deterministic environment with non-fixed scan cycles.

48. However Meek teaches a virtual CPU (virtual CPU 402, Fig. 4 (also referred to as metadata engine), col. 8, lines 18-19) adapted to provide replaceable functionality (e.g. replacement routines) to the operation of the PLC (replacement routines 400 on run on a virtual CPU 402, Fig. 4).

49. It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified the CPU of Papadopoulos with the teachings of a virtual CPU from Meek because this feature would have provided a mechanism to

execute of replacement routines (e.g. extensions), which could have been used in various different hardware platforms because they are provided in platform-independent interpretive code, on a virtual CPU which uses the same instruction set in any system platform (col. 8, lines 13-24 of Meek).

50. In addition, Kodosky teaches the system comprising an Application Wizard adapted to provide a first mode and a second mode (col. 22, lines 39-54), the first mode adapted to generate a Real Time Dynamic Link Library that is usable in a deterministic environment with fixed scan cycles (co. 22, lines 41-43), the second mode adapted to generate a Dynamic Link Library that is usable in a non-deterministic environment with non-fixed scan cycles (col. 22, lines 41-45).

51. It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have further modified the PLC's Application Program Request of Papadopoulos as modified by Meek with the teachings of a loading a DLL from Kodosky because this feature would have further provided a mechanism for an embedded application to load DLLs and to invoke or call functions in DLLs (col. 22, lines 39-41).

52. Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over Papadopoulos et al. (hereinafter Papadopoulos) (U.S. Patent 6,282,454) in view of Meek et al. (hereinafter Meek) (U.S. Patent 6,121,924), and further in view of Kodosky

et al. (hereinafter Kodosky) (U.S. Patent 6,173,438 B1) as applied to claim 17 above, and further in view of Hunt et al. (hereinafter Hunt) (U.S. Patent 6,539,422 B1).

53. As to claim 18, Papadopoulos as further modified by Kodosky teaches the invention substantially as claimed including teaches the system of claim 17, further comprising:

a system block loader (back plane driver 56, Fig. 3) adapted to load system blocks (back plane driver 56, Fig. 3 receives request from PLC's ladder logic MSTR blocks, col. 5, lines 29-30 of Papadopoulos), the system blocks including at least one of a system function block, a system function, a system data block (MSTR blocks are functions which include reading and writing data and allow programs running on the PLC to send commands and receive responses, col. 5, lines 22-28 of Papadopoulos).

54. Papadopoulos as further modified by Kodosky does not explicitly disclose an ODK SB Add-on dynamic link library (DLL) for implementing a common object module (COM) interface for the virtual CPU and system block loader.

55. However Hunt teaches an ODK SB Add-on dynamic link library (DLL) for implementing a common object module (COM) interface for the virtual CPU and system block loader (col. 9, lines 65-67 of Hunt).

56. It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have further modified the back plane driver of Papadopoulos as further modified by Kodosky with the teachings of a DLL for implementing a COM interface from Hunt because this feature would have further provided a separate file a programmer may make connections to a module (e.g. client, server, HTTP task) without effecting the operation of the calling program or any other routine (col. 10, lines 5-8 of Hunt).

57. Claim 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over Papadopoulos et al. (hereinafter Papadopoulos) (U.S. Patent 6,282,454) in view of Meek et al. (hereinafter Meek) (U.S. Patent 6,121,924), and further in view of Kodosky et al. (hereinafter Kodosky) (U.S. Patent 6,173,438 B1) as applied to claim 17 above, and further in view of Hammond (U.S. Patent 5,974,470).

58. As to claim 19, Papadopoulos as further modified by Kodosky does not explicitly disclose wherein the one or more extensions are dynamic link libraries.

59. However Hammond teaches the system of claim 17, wherein the one or more extensions are dynamic link libraries (DLL consists of functions any application can use (e.g. extends the application functionality), col. 1, lines 33-34 of Hammond).

60. It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have further modified the web server client, server, and HTTP tasks of Papadopoulos as further modified by Kodosky with the teachings of a dynamic link library from Hammond because this feature would have further provided patches to standard Windows™ API call logic with code that serves to more accurately load the correct DLL's associated with specified applications (col. 3, lines 51-56, of Hammond).

Response to Arguments

61. Applicant's arguments with respect to claims 1-38 have been considered but are moot in view of the new ground(s) of rejection.

Claim Objections

62. The combination of claims 6, 10, 13, and claim 17, lines 4-6 and 12-13 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Conclusion

63. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

6715139 B1 to Kodosky et al. and 7,340, 717 B2 disclose generating and loading DLLs in a real-time environment.

64. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

65. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

66. Any inquiry concerning this communication or earlier communications from the examiner should be directed to KimbleAnn Verdi whose telephone number is (571)270-1654. The examiner can normally be reached on Monday-Friday 7:30am 5:00pm EST..

67. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571 273-8300.

68. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Meng-Ai An/
Supervisory Patent Examiner, Art Unit 2195

KV
June 24, 2008